CLAIMS

WE CLAIM:

1. A method for processing media data, the method comprising:

receiving one or more media data streams via a control layer;

modifying the data streams in one or more stream sinks;

implementing in a media sink one or more state machines to control a state of transfer of the media data streams, the media sink responding to the control layer; and

using the state of the media data streams to modify the functionality of the stream sinks.

2. The method of claim 1 wherein the modification of the data streams is dynamic.

3. The method of claim 1 further comprising:

throttling the processing via the stream sinks based on one or more media sink components.

4. The method of claim 1 wherein one or more of the media sink and the stream sinks provide notifications for events to the control layer.

5. The method of claim 1 wherein the media data stream switches to a second media sink upon a detection of invalid media sink.

6. The method of claim 1 wherein the media sink directs multiplexing of two or more of the media data streams into a same media sink.

7. The method of claim 1 wherein the control layer directs control and timing for the media sink and the stream sinks.

8. The method of claim 1 wherein the control layer directs format negotiation to be performed in the stream sinks, the format appropriate for an output device.

9. The method of claim 1 wherein the control layer includes a media engine and a media processor, the media engine communicating with a core layer to direct a pipeline through one or more multimedia transforms and to the media sink.

10. The method of claim 9 wherein the core layer includes the media sink, stream sinks a media source, the multimedia transforms and stream sources.

11. The method of claim 1 wherein the stream sink accesses an application programming interfaces (API) that enables the stream sink to access a pointer to the media sink.

12. The method of claim 1 wherein the stream sink accesses an application programming interfaces (API) that provides an identifier for the media sink.

13. The method of claim 1 wherein the stream sink accesses an application programming interfaces (API) that provides a type of media in use.

14. The method of claim 1 wherein the stream sink accesses an application programming interfaces (API) configured to cause processing of a sample of the media data.

15. The method of claim 1 wherein the stream sink accesses an application programming interfaces (API) configured to remove any data that has not been processed.

16. The method of claim 1 wherein the stream sink accesses an application programming interfaces (API) configured to place a marker in the data stream to determine when the stream sink has finished processing received data associated with the marker.

17. The method of claim 1 wherein the stream sink accesses an application programming interfaces (API) configured to identify an end of a segment of the media data.

18. The method of claim 1 wherein the core layer is configured to communicate to retrieve characteristics of a sample allocator.

19. The method of claim 1 wherein the core layer is configured to request that a sample allocator acquire any needed resources.

20. The method of claim 1 wherein the core layer is configured to request that a sample allocator end an asynchronous resource allocation process.

21. The method of claim 1 wherein the core layer is configured to request that a sample allocator retrieve one or more of a maximum number of samples in a sample allocation and any requested samples.

22. The method of claim 1 wherein the core layer is configured to request that a sample allocator cancel one or more allocations.

23. A computer readable medium having computer-executable instructions for processing data through a collection of one or more media objects, the computer-executable instructions performing acts comprising:

receiving one or more media data streams via a control layer;

modifying the data streams in one or more stream sinks;

implementing in a media sink one or more state machines to control a state of transfer of the media data streams, the media sink responding to the control layer; and

using the state of the media data streams to modify the functionality of the stream sinks.

24. The computer readable medium of claim 23 wherein the modification of the data streams is dynamic.

25. The computer readable medium of claim 23 wherein the acts are further comprising: throttling the processing via the stream sinks based on one or more media sink components.

26. The computer readable medium of claim 23 wherein one or more of the media sink and the stream sinks provide notifications for events to the control layer.

27. The computer readable medium of claim 23 wherein the media data stream switches to a second media sink upon a detection of invalid media sink.

28. The computer readable medium of claim 23 wherein the media sink directs multiplexing of two or more of the media data streams into a same media sink.

29. The computer readable medium of claim 23 wherein the control layer directs control and timing for the media sink and the stream sinks.

30. The computer readable medium of claim 23 wherein the control layer directs format negotiation to be performed in the stream sinks, the format appropriate for an output device.

31. The computer readable medium of claim 23 wherein the control layer includes a media engine and a media processor, the media engine communicating with a core layer to direct a pipeline through one or more multimedia transforms and to the media sink.

32. A multimedia system comprising:

a control layer configured to receive one or more media data streams from an application;

a core layer coupled to the control layer, the core layer including:

one or media sink components configured to implement one or more state machines to control transfer of the media data streams through the multimedia system; and

one or more stream sinks configured to dynamically modify the media data streams via the control layer and an identified state of the media data streams determined in the media sink components.

33. The multimedia system of claim 32 wherein the control layer is an application programming interface (API).

34. The multimedia system of claim 32 wherein the control layer includes a media engine and a media processor, the media engine communicating with a core layer to direct a pipeline through one or more multimedia transforms and to the media sink.

35. The multimedia system of claim 32 wherein the core layer includes the media sink, the stream sinks, a media source, one or more multimedia transforms and one or more stream sources.

36. The multimedia system of claim 32 wherein the core layer is configured to communicate with the media sink to retrieve the characteristics of the media sink.

37. The multimedia system of claim 32 wherein the core layer is configured to communicate with the media sink to add an additional stream sink and remove one of the stream sinks.

38. The multimedia system of claim 32 wherein the core layer is configured to communicate with the media sink, the media sink enabled to report the number of stream sinks associated with a given media sink.

39. The multimedia system of claim 32 wherein the core layer is configured to communicate with the stream sinks to send a pointer to a stream sink associated with the media sink by an index in the media sink.

40. The multimedia system of claim 32 wherein the core layer is configured to communicate to send a pointer to a stream sink associated with the media sink using a stream sink identifier.

41. The multimedia system of claim 32 wherein the core layer is configured to communicate to set a rate of a presentation clock and retrieve a presentation clock setting.

42. The multimedia system of claim 32 wherein the core layer is configured to communicate to retrieve characteristics of a sample allocator.

43. The multimedia system of claim 32 wherein the core layer is configured to request that a sample allocator acquire any needed resources.

44. The multimedia system of claim 32 wherein the core layer is configured to request that a sample allocator end an asynchronous resource allocation process.

45. The multimedia system of claim 32 wherein the core layer is configured to request that a sample allocator retrieve one or more of a maximum number of samples in a sample allocation and any requested samples.

46. The multimedia system of claim 32 wherein the core layer is configured to request that a sample allocator cancel one or more allocations.